REMARKS

The examiner is thanked for the performance of a thorough search.

By this amendment, Claim 1 is amended. No claims have been cancelled or added. Hence, Claims 1-4, 6-13, 15-20, 39-42, 44-51, and 53-58 are pending in the application.

Each issue raised in the Office Action mailed February 7, 2007 is addressed hereinafter.

I. SUMMARY OF THE REJECTIONS/OBJECTIONS

Claim 1 is objected to because of a minor formality, which is corrected by the amendment made herein.

Claims 1-4, 6-13, 15-17, and 56-58 stand rejected under 35 U.S.C. § 101 as allegedly being directed to non-statutory subject matter. This rejection is respectfully traversed.

Claims 1-4, 7-11, 13, 16-20, 39-42, 45-49, 51, and 54-58 stand rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over U.S. Patent No. 6,950,822 issued to Idicula et al. ("*Idicula*") in view of U.S. Patent No. 5,911,143 issued to Deinhart et al. ("*Deinhart*"). This rejection is respectfully traversed.

Claims 6, 15, 44, and 53 stand rejected under 35 U.S.C. 103(a) as allegedly being unpatentable over *Idicula* in view of *Deinhart*, and further in view of U.S. Patent No. 6,233,576 issued to Lewis ("*Lewis*"). This rejection is respectfully traversed.

Claims 12 and 50 stand rejected under 35 U.S.C. 103(a) as allegedly being unpatentable over *Idicula* in view of *Deinhart*, and further in view of Official Notice. This rejection is respectfully traversed.

II. ISSUES NOT RELATING TO THE CITED ART

Claims 1-4, 6-13, 15-17, and 56-58 stand rejected under 35 U.S.C. § 101 as allegedly being directed to non-statutory subject matter. Specifically, page 3 of the Office Action asserts:

> both the apparatus and method claims may be considered to be software, per se, since both claims fail to be integrated into a computer hardware system for execution. Therefore, since the claims simply recite but simple recite [*sic*] sections and steps of implementation, said claims constitute non-statutory subject matter since they **fail to fall within a statutory category**. (emphasis added)

This is incorrect. Method Claims 1-4, 6-13, and 15-17 recite a **process**, which is one of the statutory categories under 35 U.S.C. § 101. Each of apparatus Claims 56-58 recite a **manufacture**, which is another one of the statutory categories under 35 U.S.C. § 101. The fact that example embodiments described in the specification *may be* implemented in software is irrelevant because each of the apparatus and method claims fall within one of the statutory classes. Further, the Office Action cites no legal authority for its position, which cannot reasonably flow from the statutory language. Based on the foregoing, removal of the 35 U.S.C. § 101 rejection with respect to Claims 1-4, 6-13, 15-17, and 56-58 is respectfully requested.

The Office Action additionally alleged that Claims 6-7 and 15-16 produce "no useful, concrete, and tangible result" because they each recite performing a certain step "only if" a certain condition is satisfied. However, because the claims upon which Claims 6-7 and 15-16 depend (Claims 1 and 10, respectively) produce at least one "useful, concrete, and tangible result", Claims 6-7 and 15-16, by virtue of their dependency, also produce that same "useful, concrete, and tangible result." Further, Claims 6-7 and 15-16 recite performing specific tangible actions including granting or denying user access under specified conditions. What happens under other conditions is irrelevant and there is no statutory or judicial requirement to recite every possible functional permutation in a claim.

III. ISSUES RELATING TO THE CITED ART

Claims 1-4, 7-11, 13, 16-20, 39-42, 45-49, 51, and 54-58 stand rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over *Idicula* in view of *Deinhart*.

A. CLAIM 1

Claim 1 recites:

> A method for controlling access to a resource, the method comprising the steps of:
> **creating and storing in a filesystem of an Operating System a file that represents the resource**;
> receiving user-identifying information from a user requesting access to the resource, wherein the user-identifying information comprises a role associated with the user, wherein the role is determined from a user identifier uniquely associated with the user and from a group identifier associated with a group that includes the user;
> receiving a resource identifier associated with the resource;
> **creating an access identifier based on the user-identifying information and the resource identifier**, **wherein the access identifier is formatted as a file attribute that is used by the Operating System to manage file access**;
> **calling the Operating System to perform a file operation on the file** by providing the access identifier to the Operating System to attempt access to the file; and
> **granting the user access to the resource when the Operating System call successfully performs the file operation**;
> wherein the file operation on the file representing the resource is selected from a group consisting of opening the file, closing the file, deleting the file, reading from the file, writing to the file, executing the file, appending to the file, reading a file attribute, and writing a file attribute. (emphasis added)

In an embodiment, resources may be a software application, software application message, server, router, switch, file, directory, etc. (Specification, paragraphs 27 and 28). A file is created and stored in a filesystem of an Operating System. The file represents a particular resource. Subsequently, user-identifying information is received from a user who is requesting access to that resource. A resource ID associated with the resource is also received, e.g., from the user. Subsequently, an access ID is created based on the user-identifying information and the resource ID. The access ID is formatted as a file attribute that is used by the Operating System to gain access to the file. The Operating System is then called to perform a file operation (e.g., open, write, delete) on the file that represents the resource. This is done by providing the access ID to the Operating System to attempt access to the file. The user is granted access to the particular resource when the Operating System call successfully performs the file operation.

Because Operating Systems calls are used to attempt access to a file, a complex security mechanism is avoided and application software developers are not required to learn a specific API and to write code against it (see Specification, paragraph 3). Claim 1 uses the filesystem of an Operating System (OS) to manage access to files. The OS filesystem is one of the most well-known and easy to write code systems in a computer system (see Specification, paragraphs 4, 7, and 8).

In contrast, *Idicula* does not teach anything about (1) an Operating System, (2) a filesystem, or (3) a file that represents the resource. Fundamentally, *Idicula* is not concerned about **controlling access to a resource**. Instead, *Idicula* teaches a technique for servicing requests for database services (Abstract). Specifically, *Idicula* teaches that rather than deleting session objects associated with a particular session when the corresponding connection ends, some of those session objects are saved and reused with new connections (col. 6, lines 6-9). Thus, the resources that would be consumed to delete and recreate session objects when connections are ended and started, respectively, are conserved (col. 6, lines 11-15).

Due to these significant differences, *Idicula* fails to teach or suggest numerous features of Claim 1.

1. Idicula *fails to teach or suggest "creating and storing in a filesystem of an Operating System a file that represents the resource"*

The Office Action cites col. 4, lines 42-56 of *Idicula* for disclosing "creating and storing in a filesystem of an Operating System a file that represents the resource", as recited in Claim 1. This is incorrect. Col. 4, lines 42-56 merely state:

> The database server 110 includes memory 120 on the database server host computer, which is allocated for use by the database server 110. The database server 110 maintains several data structures in memory 120. Among the data structures maintained in memory 120 are zero or more **session objects 122, one or more process state objects 130a, 130b, collectively referenced hereinafter as process state objects 130, and a**

> **session pool object 140. In object-oriented technologies, an object is a data structure that stores data that indicates one or more attributes or methods or both**. An object is one instance of an object type that is defined in one or more object type data structures. In other embodiments, the data structures that are illustrated as objects in FIG. 1 need not be objects according to object-oriented technologies. (emphasis added based on Office Action).

This cited portion of *Idicula* lacks any teaching or suggestion of <u>creating a file</u> or storing a file <u>in a filesystem</u>. Furthermore, there is no concept in the cited portion of a file, **much less a file that represents a resource**. At best, session objects 122 represent a session, which is "a related series of one or more requests for services made over a communication channel" (col. 1, lines 57-58). However, **session object 122 cannot be a file**. Session objects 122 are <u>not</u> stored in a filesystem, as Claim 1 would require. Instead, session objects 122 reside only in memory.

In another possible interpretation, a session object 122 may be considered a resource. However, there is nothing in *Idicula* that <u>represents</u> a session object 122. Furthermore, a user in *Idicula* is <u>not</u> "requesting access to" a session object 122, as Claim 1 would require.

2. Idicula *fails to teach or suggest "creating an access identifier..., wherein the access identifier is formatted as a file attribute that is used by the Operating System to manage file access"*

The Office Action cites col. 4, lines 42-56 of *Idicula* (the same portion cited above) for disclosing "creating an access identifier..., wherein the access identifier is formatted as a file attribute that is used by the Operating System to manage file access", as recited in Claim 1. This is incorrect. Col. 4, lines 42-56 merely states, among other things:

> ...session objects 122, one or more process state objects 130a, 130b, collectively referenced hereinafter as process state objects 130, and a session pool object 140. In object-oriented technologies, an object is a data structure that stores data that indicates one or more attributes or methods or both.

Applicants are not clear as to what in this cited portion of *Idicula* the Office Action is equating to the "access identifier" of Claim 1. At best, the Office Action is equating one of session objects

122, a process state object 130, or session pool object 140 of *Idicula* to the "access identifier" of Claim 1. However, none of these elements of *Idicula* are **formatted as a file attribute**. Furthermore, none of these elements of *Idicula* are used by an Operating System to **manage file access**.

Additionally, Claim 1 states that the access identifier is **created based on** user-identifying information and **a resource identifier**. The Office Action cites "If a session is already created for this client, a session object 122 associated with the client is indicated in the process state object 130" of *Idicula* for disclosing "receiving a resource identifier associated with the resource", as recited in Claim 1 (page 4). It appears that the Office Action is equating the session object 122 of *Idicula* with the "resource identifier" of Claim 1. If that is true, then it is unclear what in *Idicula* is **created based on the session object 122**, as Claim 1 would require.

3. Idicula *fails to teach or suggest "calling the Operating System to perform a file operation on the file by providing the access identifier to the Operating System to attempt access o the file"*

The Office Action cites col. 1, lines 52-62 and col. 7, lines 19-30 of *Idicula* for disclosing "calling the Operating System to perform a file operation on the file by providing the access identifier to the Operating System to attempt access to the file", as recited in Claim 1. This is incorrect. Col. 1, lines 52-62 merely states, among other things:

> A session is a related series of one or more requests for services made over a communication channel. The channel is typically established by the operating system of the host for the database server....

Col. 7, lines 19-30 merely states, among other things:

> If a session is already created for this client, a session object 122 associated with the client is indicated in the process state object 130; and that session object 122 is used.

These cited portions of *Idicula* fail to teach or suggest performing file operations on a file or providing an access identifier to an Operating System. It is not even clear what in *Idicula* the Office Action is equating with "file operation", "file", or "access identifier" of Claim 1

4. Idicula *fails to teach or suggest "granting the user access to the resource when the Operating System call successfully performs the file operation"*

The Office Action cites col. 7, lines 20-21 of *Idicula* for disclosing "granting the user access to the resource when the Operating System call successfully performs the file operation", as recited in Claim 1. This is incorrect. Col. 7, lines 20-21 merely states:

> For example, a request is received from database client 102a for database services.

It is unclear what is **granted** in this portion of *Idicula.* Furthermore, there is no concept of successfully performing a **file operation**, much less an Operating System performing a file operation.

Based on the foregoing, *Idicula* fails to teach or suggest numerous features of Claim 1. Additionally, *Dienhart* is not used to show the features of Claim 1 that are not taught or suggested by *Idicula.* Therefore, Claim 1 is patentable over *Idicula* and *Deinhart.* Removal of the 35 U.S.C. § 103(a) rejection with respect to Claim 1 is therefore respectfully requested.

B. CLAIMS 10, 18, 39, 48, AND 56

The Office Action stated the same reasons in rejecting Claims 10 and 18 to those in rejecting present Claim 1. Also, Claims 39, 48 and 56 recite features discussed above that make Claim 1 patentable over *Idicula* and *Deinhart.* Therefore, for at least the same reasons set forth above by the Applicant in connection with present Claim 1, it is respectfully submitted that each of Claims 10, 18, 39, 48 and 56 is patentable over *Idicula* and *Deinhart.*

C. DEPENDENT CLAIMS

The dependent claims not discussed thus far are dependent claims, each of which depends (directly or indirectly) on one of the independent claims discussed above. Each of the dependent claims is therefore allowable for the reasons given above for the claim on which it depends. In addition, each of the dependent claims introduces one or more additional limitations that independently render it patentable. However, due to the fundamental differences already identified, to expedite the positive resolution of this case, a separate discussion of those limitations is not included at this time. The Applicant reserves the right to further point out the differences between the cited art and the novel features recited in the dependent claims.

III. CONCLUSIONS & MISCELLANEOUS

For the reasons set forth above, all of the pending claims are now in condition for allowance. The Examiner is respectfully requested to contact the undersigned by telephone relating to any issue that would advance examination of the present application.

A petition for extension of time, to the extent necessary to make this reply timely filed, is hereby made. If applicable, a law firm check for the petition for extension of time fee is enclosed herewith. If any applicable fee is missing or insufficient, throughout the pendency of this application, the Commissioner is hereby authorized to any applicable fees and to credit any overpayments to our Deposit Account No. 50-1302.

Respectfully submitted,
HICKMAN PALERMO TRUONG & BECKER LLP

Dated: May 2, 2007

/DanielDLedesma/
Daniel D. Ledesma
Reg. No. 57,181

2055 Gateway Place Suite 550
San Jose, California 95110-1093
Telephone No.: (408) 414-1229
Facsimile No.: (408) 414-1076